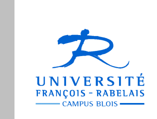
Ergonomie des Interfaces Homme-Machine

Jean-Yves Antoine

http://www.sir.blois.univ-tours.fr/~antoine/



Ergonomie des Interfaces Homme-Machine

Chapitre 2 — Facteurs humains

Quelques notions importantes sur la qualité des logiciels interactifs vue du point de vue de l'utilisateur humain et leur traduction en terme de recommandations ergonomiques

INTRODUCTION - Objectifs

2.1. Notions

2.1.1. Connaître les principaux facteurs influençant l'ergonomie des IHMs
2.1.2. Savoir relier principes ergonomiques et facteurs humains

2.2. Pratiques

2.2.1. Savoir détecter et catégoriser (facteur humain affecté et critère ergonomique non respecté) un problème dans la conception de l'IHM
2.2.2. Comprendre et savoir utiliser des directives compilant des règles ergonomiques

QUALITE D'UN LOGICIEL / SITE WWW

Le meilleur des logiciels est voué à l'échec si son IHM est mal conçue
… mais l'interface d'un logiciel interactif ne fait pas tout !

Fonctionnalités adéquates

- Analyse complète de la tâche
- Préalable à la conception de l'interface

Fiabilité / sûreté du logiciel

- **Erreurs** : principale source de défiance vis-à-vis d'un logiciel
- Travail sur la prévention et la récupération des erreurs

QUALITE D'UN LOGICIEL : UTILISABILITE

Utilisabilité d'un logiciel interactif (Dix, Finlay *et al*., 2003)

- **Apprenabilité** — facilité avec laquelle l'utilisateur peut prendre en main le logiciel et découvrir ses fonctionnalités
- **Flexibilité** — capacité du système à offrir des modes d'interactions multiples et à s'adapter
- **Robustesse** — niveau de satisfaction dans la réalisation des tâches permises par le système (ne se limite pas à la fiabilité)

Facteurs humains (perçu par l'utilisateur)	**Causes** (principes ergonomiques)
Apprenabilité	causabilité, apprenabilité, consistance / cohérence, familiarité, génériticité
Flexibilité	adaptation, styles d'utilisation
Robustesse	prévention (observabilité, causalité, guidage), gestion des erreurs

QUALITE D'UN LOGICIEL : APPRENABILITE

- **Causalité** — facilité de déterminer le comportement du système à partir d'interactions antérieures.

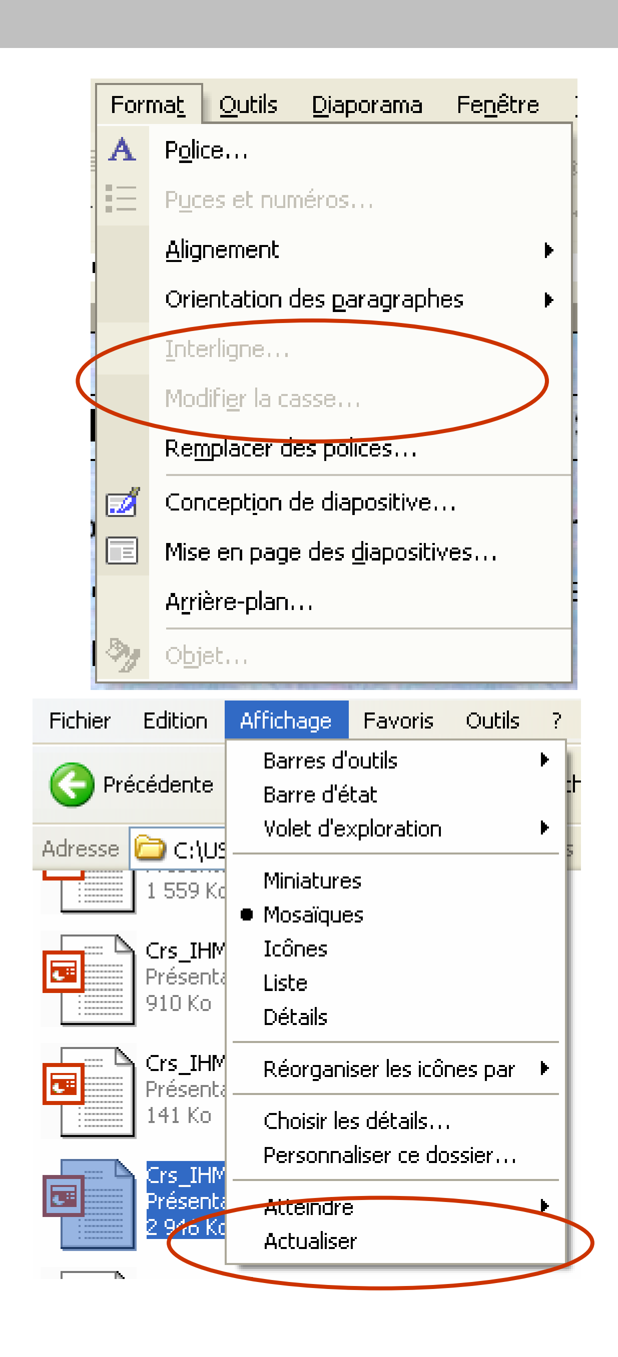
  ⑤ visibilité opérationnelle

- **Observabilité** — facilité offerte à l'utilisateur de vérifier les effets de ses actions.



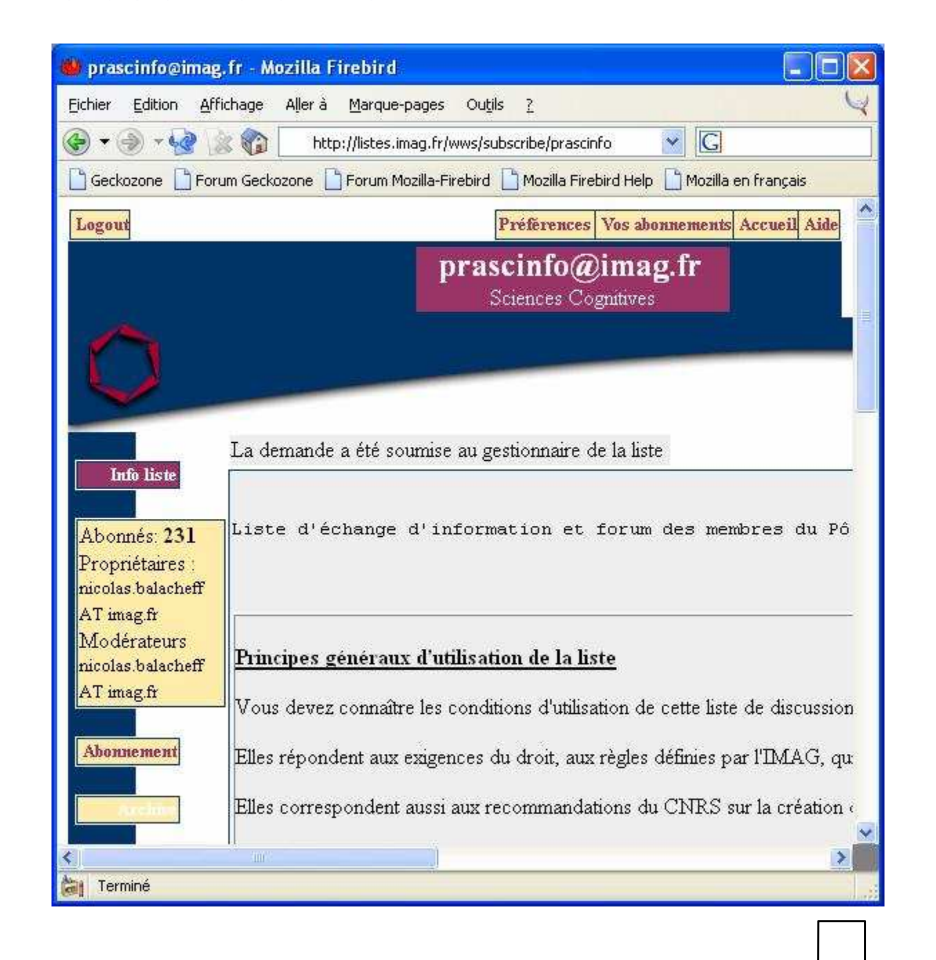
  ⑤ observabilité directe vs. indirecte

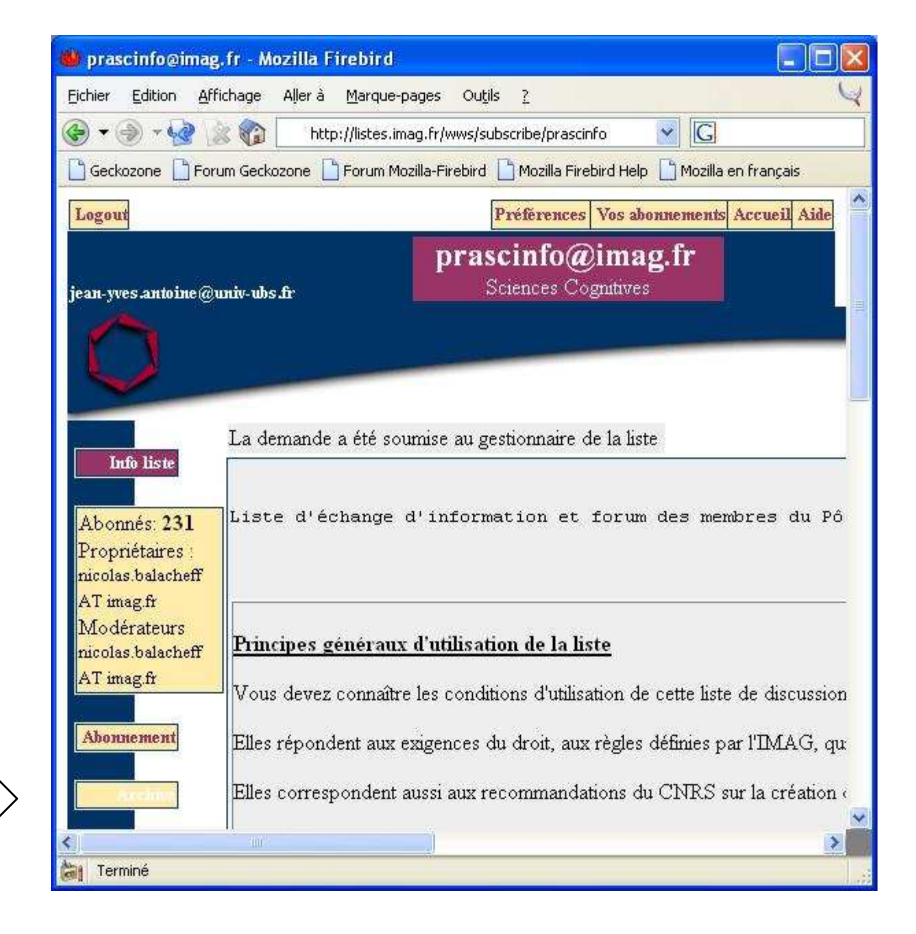
  Exemple: anciennes versions du Finder du Macintosh ou de l'explorateur Windows



QUALITE D’UN LOGICIEL : APPRENABILITE

Observabilité ?





QUALITE D'UN LOGICIEL : APPRENABILITE

Observabilité : exemple

Gestion de l'attente en cas de temps de réponse importants : informer l'utilisateur de l'avancement du travail

Temps d'attente prévu	Affichage
2 à 6 secondes	icône d'attente (sablier, horloge…)
6 à 30 secondes	Message avec marque d'avancement (% réalisé, temps d'attente)
> 30 secondes	Idem avec en outre un détail des actions en cours de réalisation

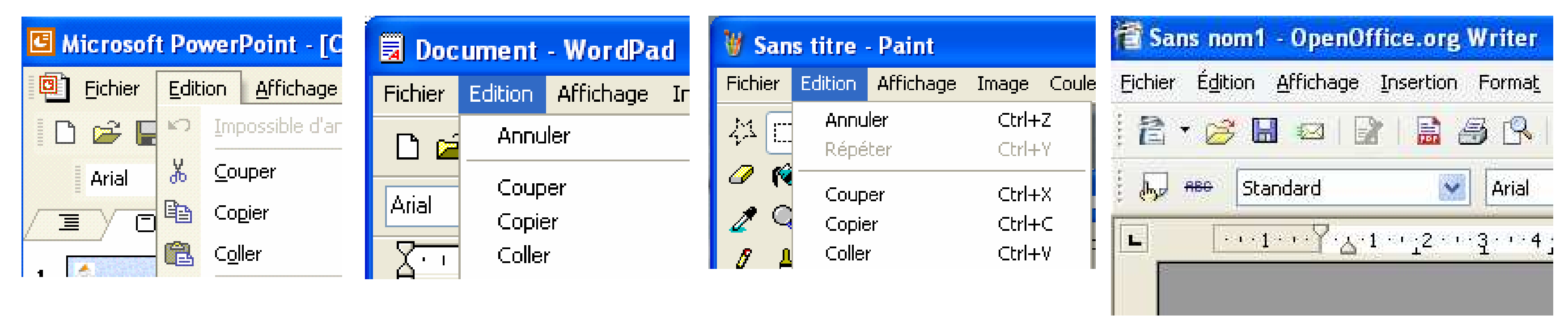


QUALITE D'UN LOGICIEL : APPRENABILITE

Consistance : facteur essentiel de qualité

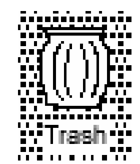
- Le comportement de l'interface doit être cohérent et homogène tout au long de l'utilisation : apprentissage facilité et utilisateur rassuré
- La consistance peut concerner toute une suite d'application

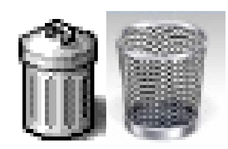
Exemple suite Microsoft Office



Contre-exemple HyperCard et MacOS (double ou simple clic)

- **Standardisation et consistance** — Certaines interfaces sont devenus des standards de facto et participent à la généricité d'une application
  - intégrer ces standard dans la conception du logiciel
  - assurer la consistance entre les versions du logiciel
  - conception attentive des fonctionnalités différentes

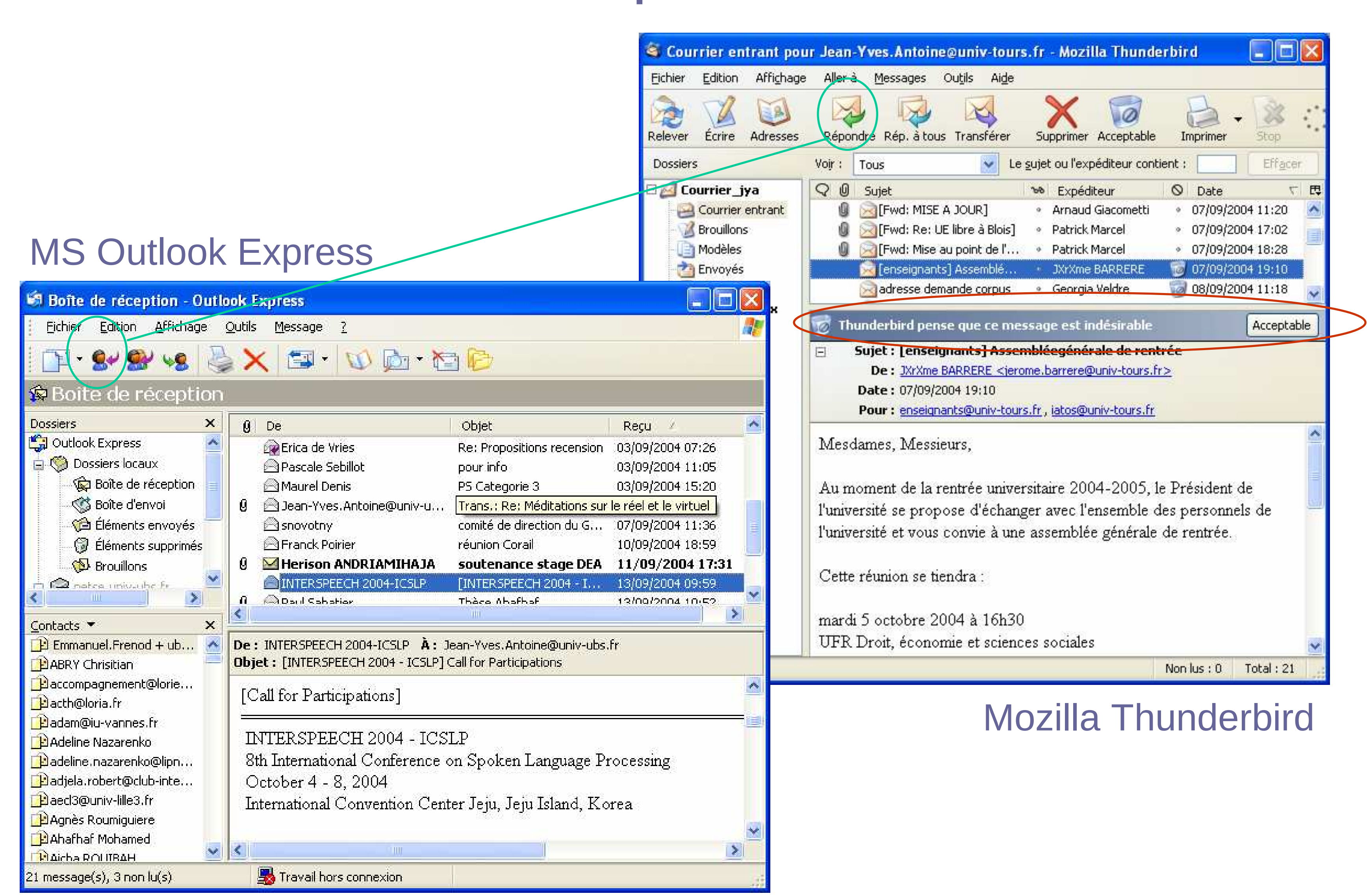






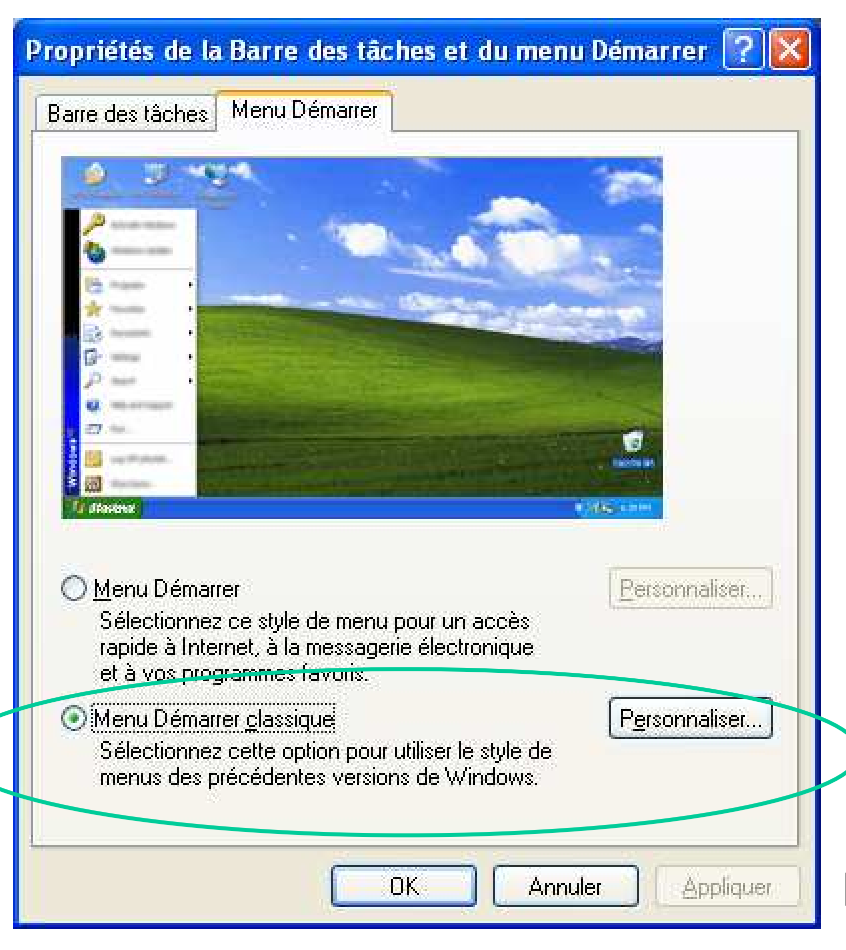
QUALITE D'UN LOGICIEL

Standardisation de facto : exemple



QUALITE D'UN LOGICIEL

Importance de la cohérence entre versions



MS Windows XP

L'utilisateur est conservateur !

QUALITE D’UN LOGICIEL : APPRENABILITE

- **Signification des codes et consistance** — Tout code (texte, icône) utilisé dans l’interface doit avoir une signification parlante et constante quel que soit le contexte d’application



contre-exemple LinuX File Manager

contre-exemple commandes Unix

`mv` « ***move*** »

⑤ historiquement, sert aussi au renommage…

`pr` « ***print*** »

⑤ formatte le fichier mais ne l’imprime pas!

`lpr` « ***l**ine **pr**int spooler* »

⑤ imprime

contre-exemple applications Windows

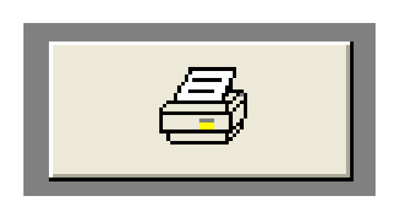


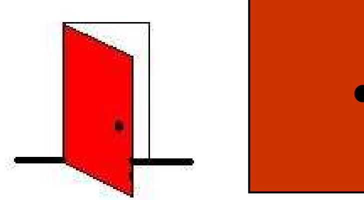
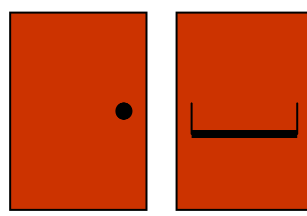
QUALITE D'UN LOGICIEL : APPRENABILITE

- **Familiarité** — similarité de l'interaction avec le monde réel ou avec l'utilisation d'autres systèmes existants.

  ⑤ métaphores : machine à écrire pour les éditeurs de texte

  ⑤ affordance (icônes)



- **Généricité** — facilité de généralisation d'expériences particulières avec le système à d'autres fonctionnalités ou d'autres logiciels : renforce le sentiment de consistance

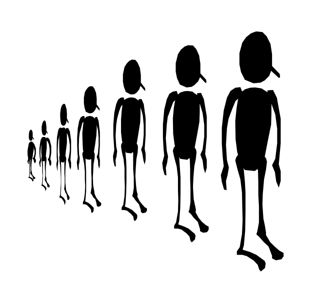
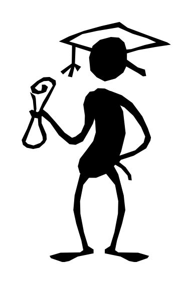


QUALITE D'UN LOGICIEL : FLEXIBILITE

Facteur essentiel : prendre en compte la diversité des utilisateurs

Dilemmes de la conception d'une IHM : *design for all*

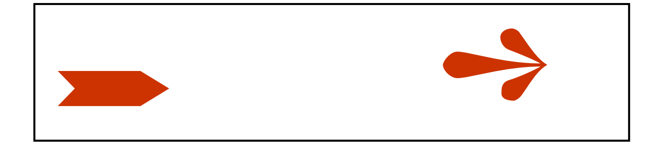
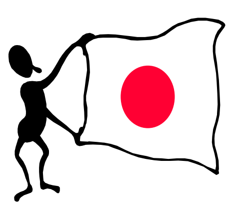
- Un logiciel s'adresse souvent à plusieurs communautés d'utilisateurs très différents




- Différences sensibles entre les cultures (*localisation*)

Exemple

DIVERSITE D'UTILISATEURS: NIVEAU D'EXPERTISE

Exemple Library of Congress (Marchinini et al., 1993)

- **Novice et primo-utilisateur** — *anxieux, à rassurer pour faciliter l'apprentissage et éviter les rejets*
  - ➢ Limiter le nombre d'actions, de concepts
  - ➢ Feedback d'information
  - ➢ Messages d'erreur informatifs
  - ➢ Manuel d'utilisation intuitif (≠ `man` !), tutoriel
  - ➢ Aide contextuelle

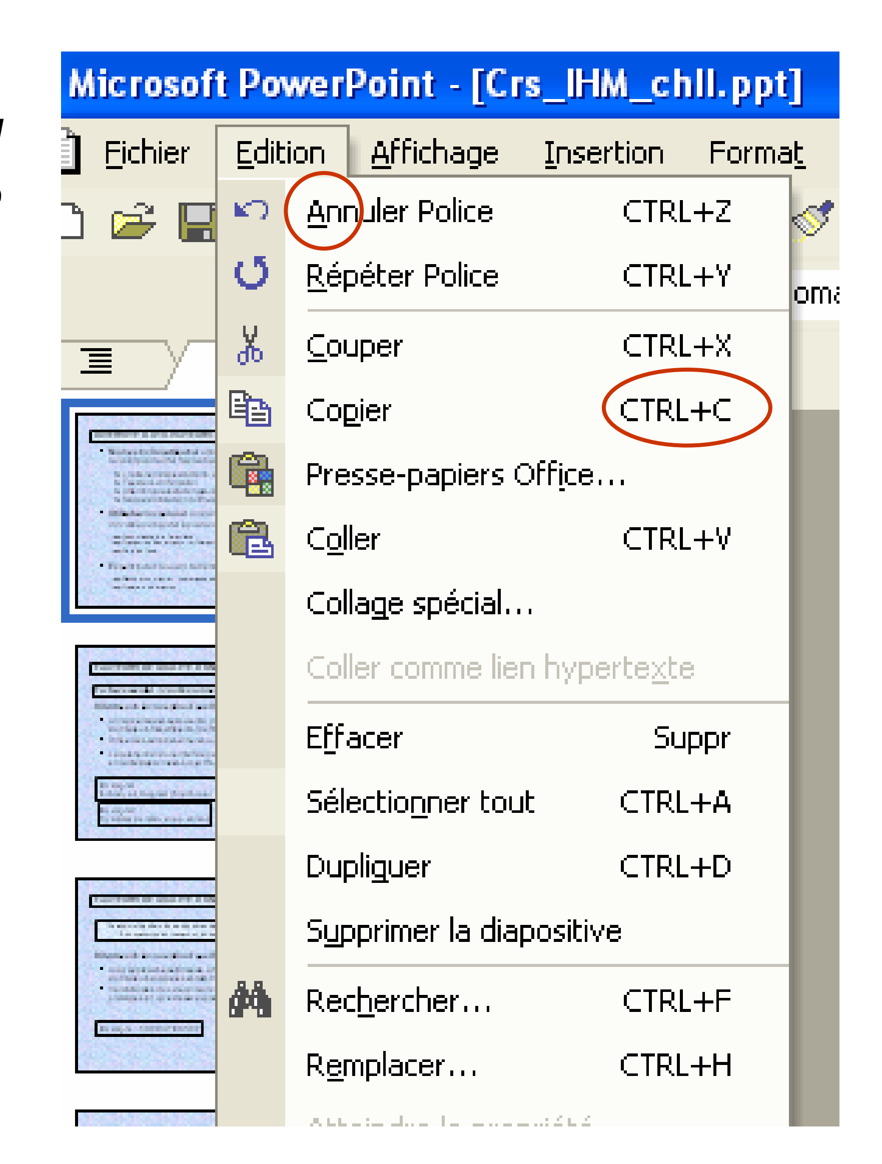


- **Utilisateur occasionnel** — *connaissance globale du système mais difficulté à se rappeler la position des différentes fonctionnalités*
  - ➢ Consistance de l'interface
  - ➢ Prévention des erreurs (utilisateur - explorateur)
  - ➢ Aide en ligne

DIVERSITE D'UTILISATEURS: NIVEAU D'EXPERTISE

- **Expert** — *excellente connaissance du domaine de la tâche, du système et de son interface: recherche avant tout l'efficacité et la rapidité*
  - Raccourcis clavier
  - Commande en ligne
  - Création de macros



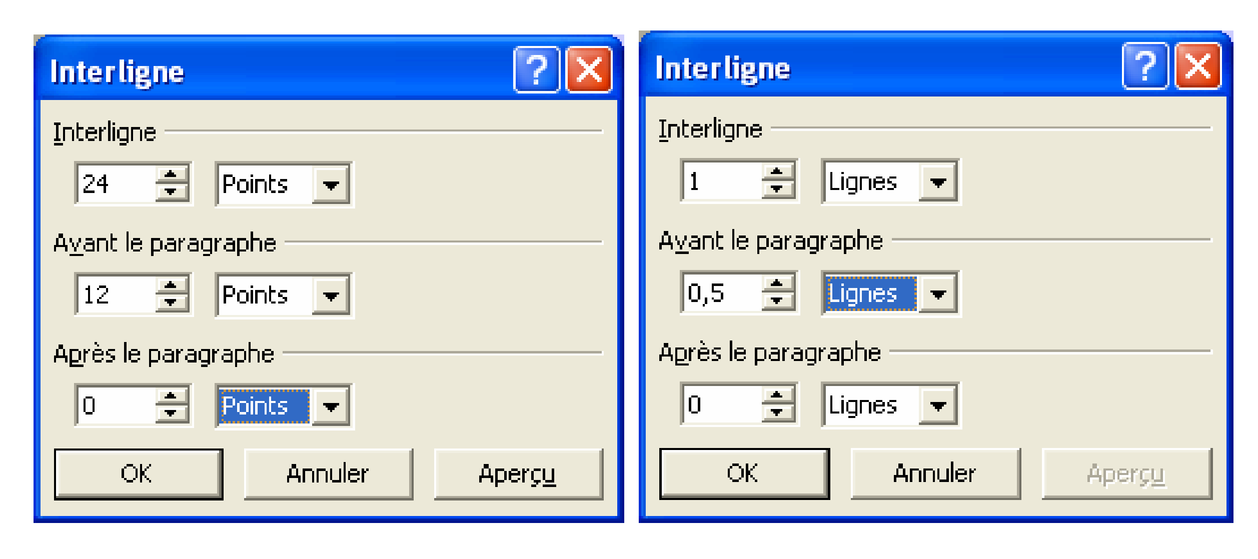


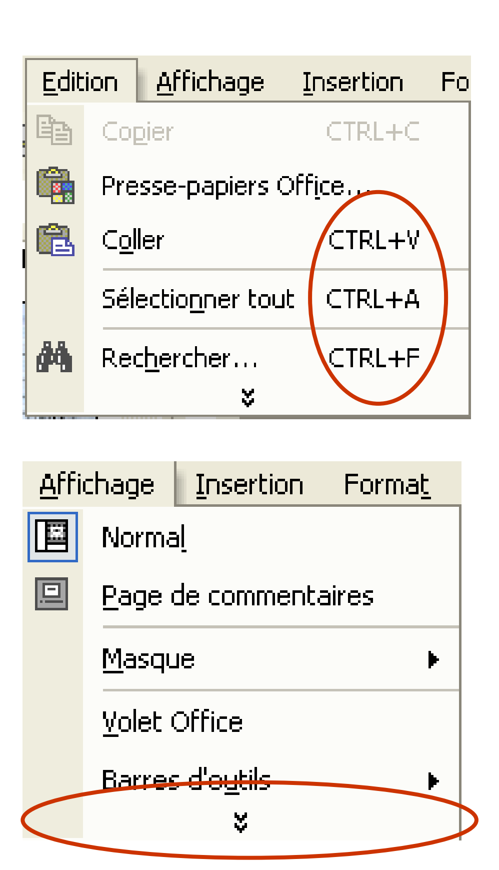
Logiciel à niveaux d'usages multiples: conception multi-niveaux

QUALITE D'UN LOGICIEL : FLEXIBILITE

- **Initiative du dialogue: contrôle utilisateur**
  - ✓ système préemptif ou non
  - ✓ transfert du contrôle entre l'utilisateur et le système

- **Adaptation**
  - ✓ **adaptatibilité** — paramétrisation par l'utilisateur
  - ✓ **adaptativité** — modification initiée par le système





Exemple: substitivité des données

QUALITE DU LOGICIEL : FLEXIBILITE

Plusieurs styles d'interaction envisageable pour une tâche donnée

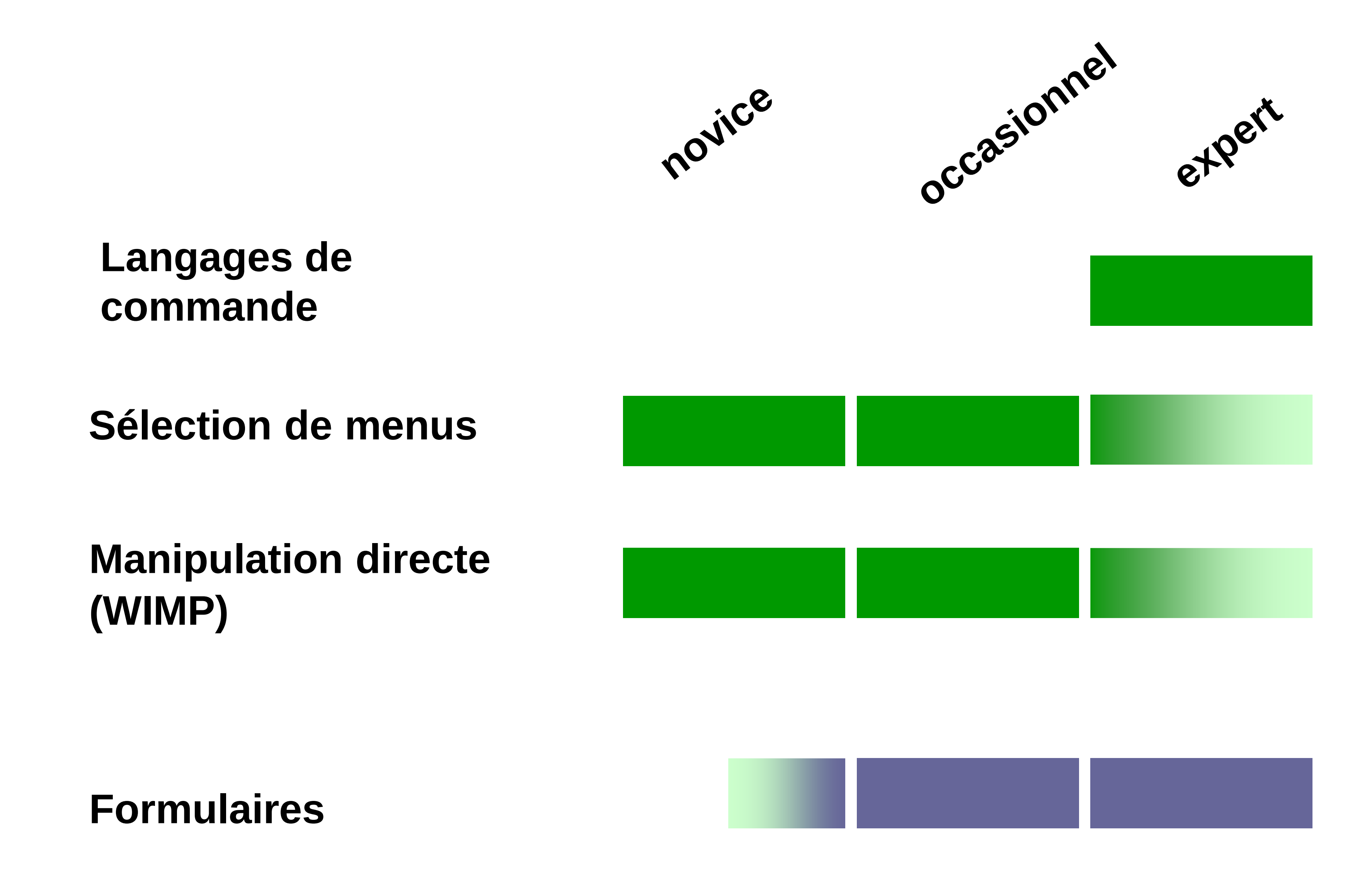
- Langage de commande
- Langages de requête et questions / réponses
- Menus
- **Manipulation directe: interfaces WIMP**
- Formulaires de saisie
- Langage naturel (écrit, parole)
- Interfaces *multitouch*, réalité augmentée…

Interfaces post-WIMP

EXEMPLE: STYLES D'INTERACTION

		
Langages de commande	**complexité** flexibilité richesse d'expression	**complexité** apprentissage erreurs
Sélection de menus	**structuration** apprentissage aisé gestion erreurs	**lenteur** masquage info peu flexible
Manipulation directe (WIMP)	**visuel** apprentissage aisé rétention apprentissage flexibilité WYSIWYG	**lenteur** (relative) pointage ambiguïté icônes coûteux en espace
Formulaires	**saisie données** apprentissage aisé prévention erreurs	usage spécifique
Langage naturel	**naturalité**	**imprécision** ambiguïté

STYLES D'INTERACTION ET NIVEAU D'EXPERTISE



QUALITE D'UN LOGICIEL : ROBUSTESSE

Rappel : fiabilité et conformité à la tâche

- ✓ complétude
- ✓ adéquation

Importance des erreurs

- Source principale d'anxiété et de rejet des utilisateurs
- Une erreur d'utilisation est aussi frustrante qu'un bug « simple »
- Influence des messages d'erreur sur l'acceptation du logiciel

Gestion et récupération des erreurs

- Rédaction essentielle des messages d'erreur : rassurer l'utilisateur et lui donner des pistes pour résoudre le problème rencontré

Prévention des erreurs

- Une conception sérieuse d'un point de vue ergonomique limitera fortement les erreurs de l'utilisateur

QUALITE D'UN LOGICIEL : ROBUSTESSE

- **Prévention des erreurs**

Observabilité et causalité — erreurs réduites si on comprend ses actions et en voit les conséquences

Ex : navigation dans un système de fichier

```
C:\USERS\JYA\IN_COGNITO>cd ASSOC

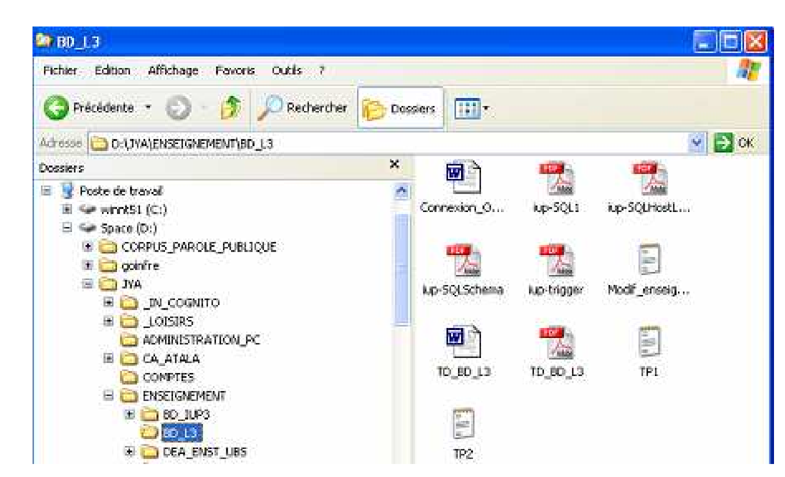
C:\USERS\JYA\IN_COGNITO\ASSOC>dir /p
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est CC19-1494

 Répertoire de C:\USERS\JYA\IN_COGNITO\ASSOC

07/09/2004  17:17    <REP>          .
07/09/2004  17:17    <REP>          ..
02/09/2004  14:48            2 974 CA_InCognito_2004.txt
18/12/2002  00:16            2 937 Rapport_moral_2002.txt
07/09/2004  17:37            6 306 Rapport_moral_2003.txt
20/11/2001  13:45           25 088 statuts_new2.1.doc
               4 fichier(s)           37 305 octets
               2 Rép(s)   3 969 785 856 octets libres

C:\USERS\JYA\IN_COGNITO\ASSOC>del CA_InCognito_2004.txt
```

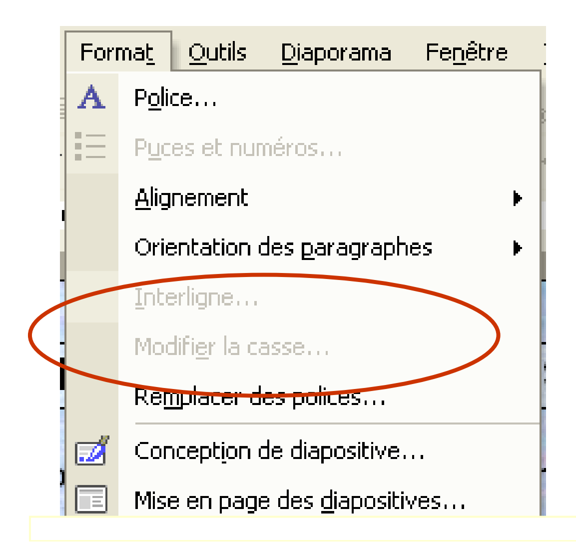
sans GUI : ligne de commande



GUI : explorateur Windows

Guidage — guider l'utilisateur (sans limiter sa liberté d'action) pour lui éviter des erreurs

- ✓ menus (option non sélectionnables grisées)
- ✓ listes dans les formulaires
- ✓ …



QUALITE D’UN LOGICIEL : ROBUSTESSE

- **Prévention des erreurs**
  - ⑤ **Limiter la charge cognitive de l’utilisateur**
  - ⑤ **Principe des efforts commensurables** — Plus une action peut avoir des conséquences graves, moins sa réalisation doit être transparente

Confirmation de la suppression du fichier

Voulez-vous vraiment envoyer 'Photos_id.doc' à la Corbeille ?

Oui Non

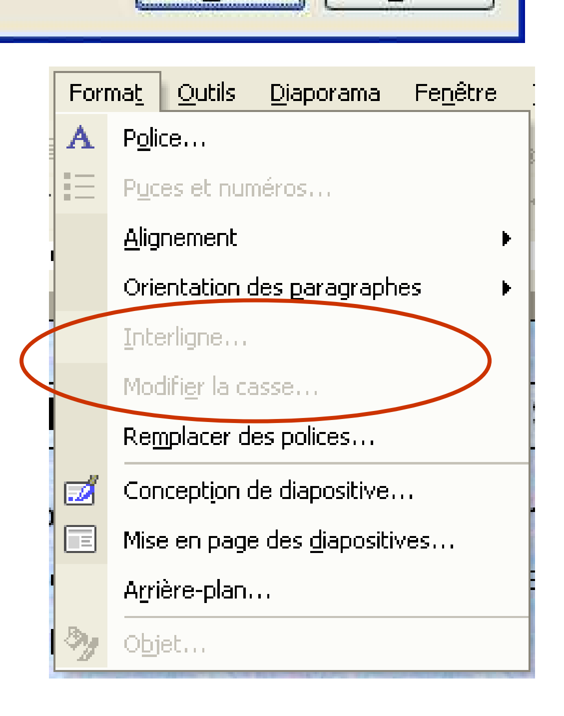
- **Prévention des erreurs : guidage**

Guider l’utilisateur (sans limiter sa liberté d’action) pour lui éviter des erreurs

✓ menus avec option non sélectionnables grisées

✓ listes dans les formulaires

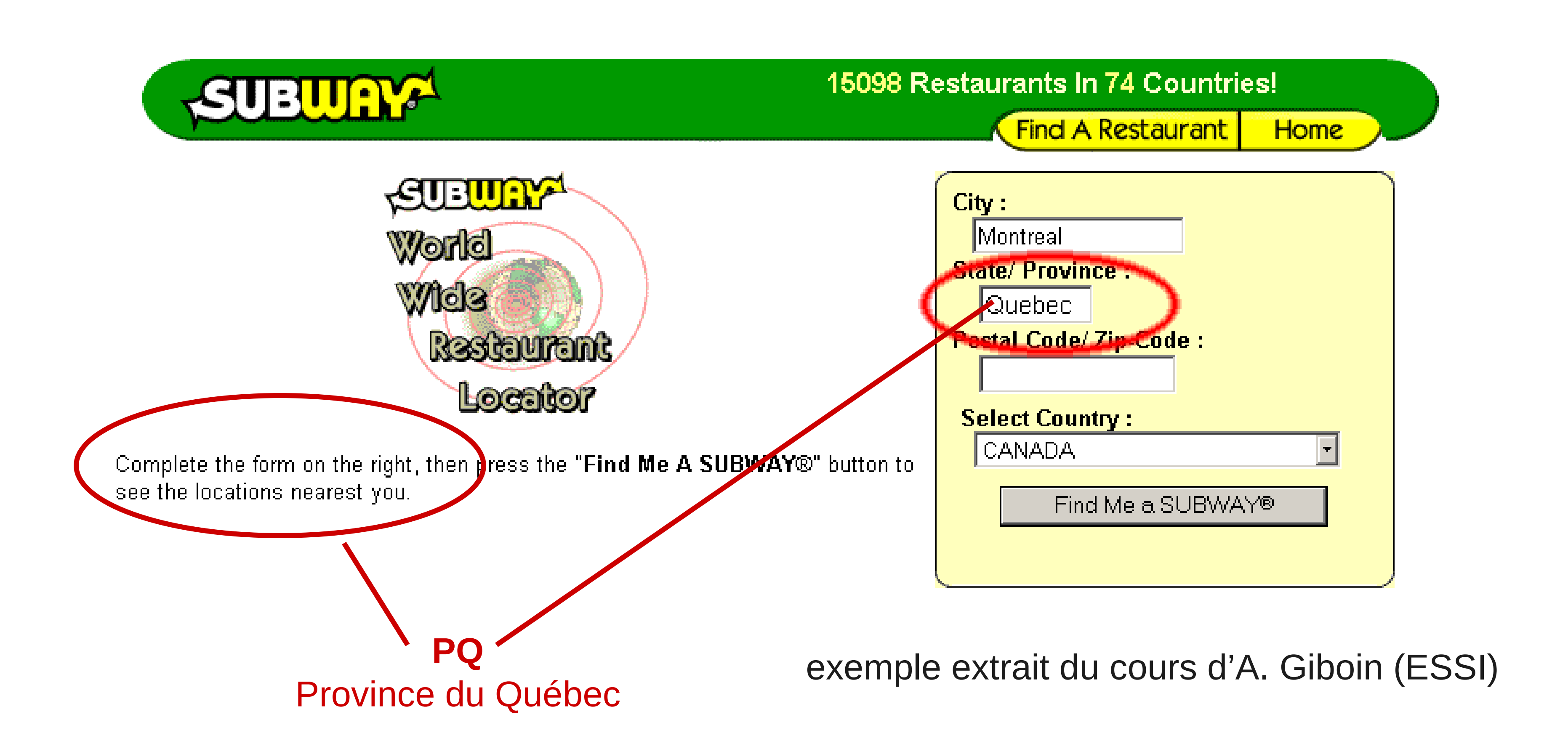
✓ …



QUALITE D'UN LOGICIEL : GUIDAGE ET ROBUSTESSE

Exemple de guidage : formulaires de saisie

- Indiquer les champs optionnels et obligatoires
- Toujours préciser le format d'une saisie
- Dire autant que possible quelles valeurs sont attendues



exemple extrait du cours d'A. Giboin (ESSI)

CONCLUSION: CONCEPTION DES IHMs

- ✓ **Multiples choix de conception** de l'IHM pour chaque fonctionnalité et chaque communauté d'utilisateurs

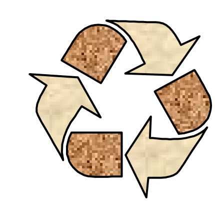
- ✓ **S'en remettre à l'intuition** du concepteur est **insuffisant**

(Contre) exemple : système réservation SOCRATE



http://perso.ens-lyon.fr/pierre.lescanne/BUGS/socrate.pdf

⑤ **Méthodologie de conception des IHM:** le cycle de vie défini en Génie Logiciel concerne également l'IHM



METHODOLOGIE DE CONCEPTION DES IHMs

- **Analyse de la tâche et des communautés d'utilisateurs** auxquelles est destiné le système
  - ⑤ modèle de la tâche (donc de l'interaction) et de l'utilisateur
  - ⑤ choix de conception de l'IHM

- **Conception : respect de principes ergonomiques**
  - ⑤ aide à déterminer les besoins (non étudié dans ce cours)
  - ⑤ oriente certains choix de conception
  - ⑤ guide l'implémentation des fonctionnalités précises

Chap. II à V

- **Conception : modèles de conception d'IHM**
- **Évaluation**
  - ⑤ Validation des choix de conception et d'implémentation par des tests d'usage sur prototypes

Chap. VI-VII

METHOLOGIE : REGLES DE CONCEPTION

Principes ergonomiques

- ✓ règles générales indépendante de la technologie mais inspirées par de la psychologie cognitive ou des sciences sociales
- ✓ règles de haut niveau
- ✓ recommandations de faible autorité (peu contraignantes)

Recommandations ou directives (*guidelines*)

- ✓ règles de conception encore assez générales mais plus liées à la technologie. Peuvent concerner des éléments précis de l'IHM
- ✓ recommandations d'autorité modérée

Standards

- ✓ règles de conception technologiques très spécifiques
- ✓ recommandations de portée limitée mais de forte autorité (contraintes explicites si on désire suivre la norme)

METHOLOGIE : REGLES DE CONCEPTION

Mise en garde

- ✓ Les directives et autres principes ergonomiques ne constituent pas des vérités absolues : confronter à la réalité du problème
- ✓ Certaines directives peuvent paraître contradictoire entre elles : bien comprendre leur background et leur champ d'application

Principe de précaution

- ✓ Les principes et autres directives sont avant tout des sources d'information issues d'expériences à prendre en considération … en en comprenant l'esprit.
- ✓ Seules certaines directives de forte autorité doivent être respectées à la lettre (norme ou respecter, standard imposé)
- ✓ Le recours à l'évaluation reste essentiel

Règles d’or de Shneiderman

- **Consistance**
- **Causalité**
- Permettre les raccourcis
- Fournir des feedbacks interactifs
- Prévenir les erreurs et faciliter leur récupération
- Autoriser l’annulation des actions
- Concision: réduire la charge cognitive de l’utilisateur (mémoire à court terme)

Principes de Norman

[Norman,1998]

I. *Use both knowledge in the world and knowledge in the head*

II. *Simplify the structure of the task*

III. *Make things visible*

IV. *Get the mappings right*

V. *Exploit the power of constraints*

VI. *Design for error*

VII. *When all else false, standardize*

Heuristiques de Nielsen

[Nielsen,1994]

I. *Visibility of system status*
II. *Match between system and real world*
III. *User control and freedom*
IV. *Consistency and standards*
V. *Error prevention*
VI. *Recognition rather than recall*
VII. *Flexibility and efficiency of use*
VIII. *Aesthetic and minimalist design*
IX. *Help users recognize, diagnose and recover from errors*
X. *Help and documentation*

PRINCIPES ERGONOMIQUES

Critères ergonomiques de Bastien et Scapin (1993)

(http://www.ergoweb.ca/criteres.html)

Norme AFNOR Z67-133-1

- ✓ Guidage
- ✓ Charge de travail
- ✓ Contrôle utilisateur explicite
- ✓ Flexibilité (⇨ *adaptabilité*)
- ✓ Gestion des erreurs
- ✓ Homogénéité (⇨ *cohérence*)
- ✓ Signifiance des Codes et Dénominations
- ✓ Compatibilité (⇨ *familiarité*)

DIRECTIVES (*GUIDELINES*)

Directives de recommandations générales

Synthèses de recommandations générales sur toutes plateformes

- ✓ Smith & Mosier (1986)
- ✓ Brown (1988)
- ✓ Mayhew (1992)
- ✓ Vanderdonkt J. (1995)

ENTRÉE DE DONNÉES

Formats des données **1.4** ***Délimiter les champs d'entrée de données***

Afin de délimiter clairement chaque champ de données, afficher des caractères spéciaux ou autres formes de mise en évidence.

EXEMPLE : Un trait de soulignement peut être utilisé à cette fin, voire un trait discontinu permettant d'indiquer le nombre de symboles requis à l'entrée tel que :

(bon) Entrer votre code : _ _ _ _ _ _
(mauvais) Entrer votre code :

EXEMPLE : [Voir la copie d'écran à la fin de cette section]

COMMENTAIRE : Une telle incitation permet de réduire les erreurs lors de la saisie de données par l'utilisateur

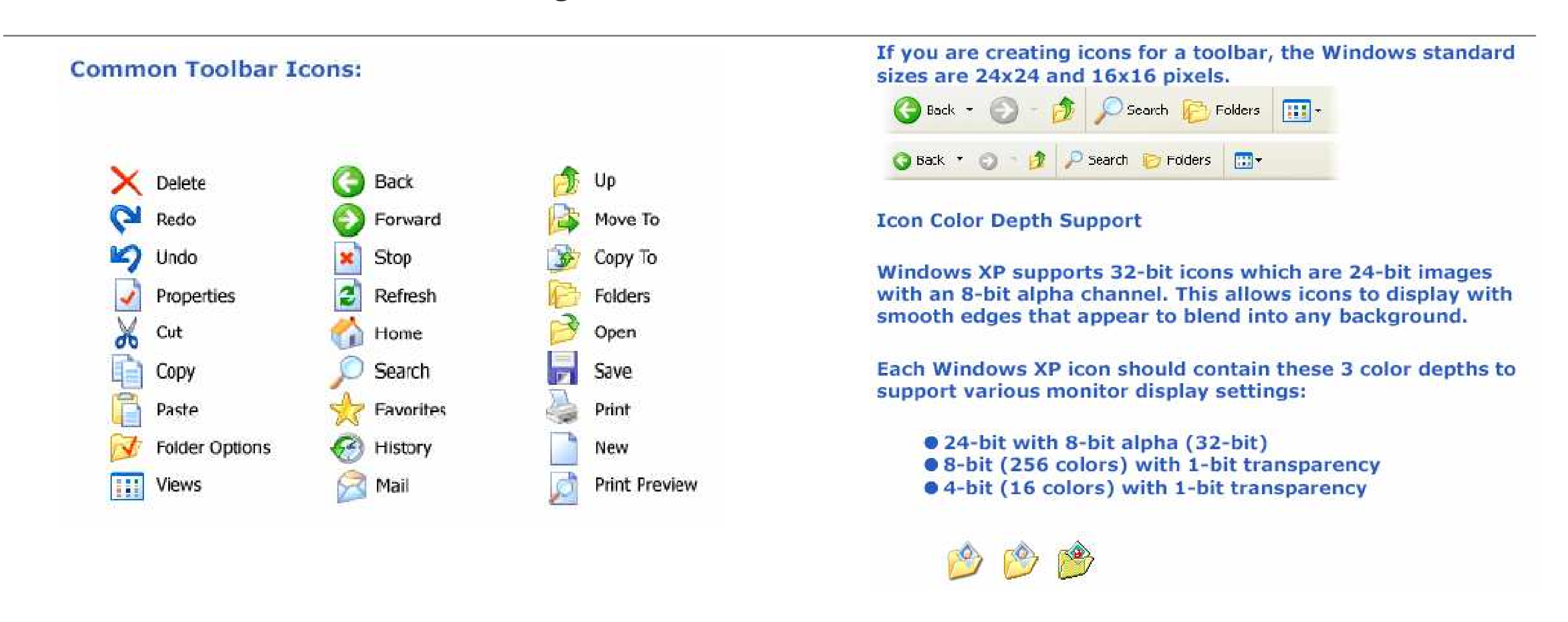
RÉFÉRENCE : BB 2.2.1; EG 6.3, 6.3.1; MS 5.15.4.3.4; PR 4.8.1; Savage, Habinek et Blackstad, 1982.

DIRECTIVES (*GUIDELINES*)

Guidelines spécifiques à une plate forme ou un GUI

Recommandations (souvent) plus concrètes permettant de favoriser, entre autres, la standardisation entres applications

- ✓ *Apple's Human Interface Guidelines*
- ✓ *Microsoft Windows XP Visual Guidelines* (id. *Vista*)
- ✓ *IBM Common User Access Advanced Interface Design Guide*
- ✓ *OSF/Motif Style Guide*
- ✓ *Java Look and Feel design Guidelines*



STANDARDS



Normes ISO (TC 159 – SC4 Ergonomie IHMs) (Blanchard, 1997)

- ✓ ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals* (*VDTs*) — *Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV)*

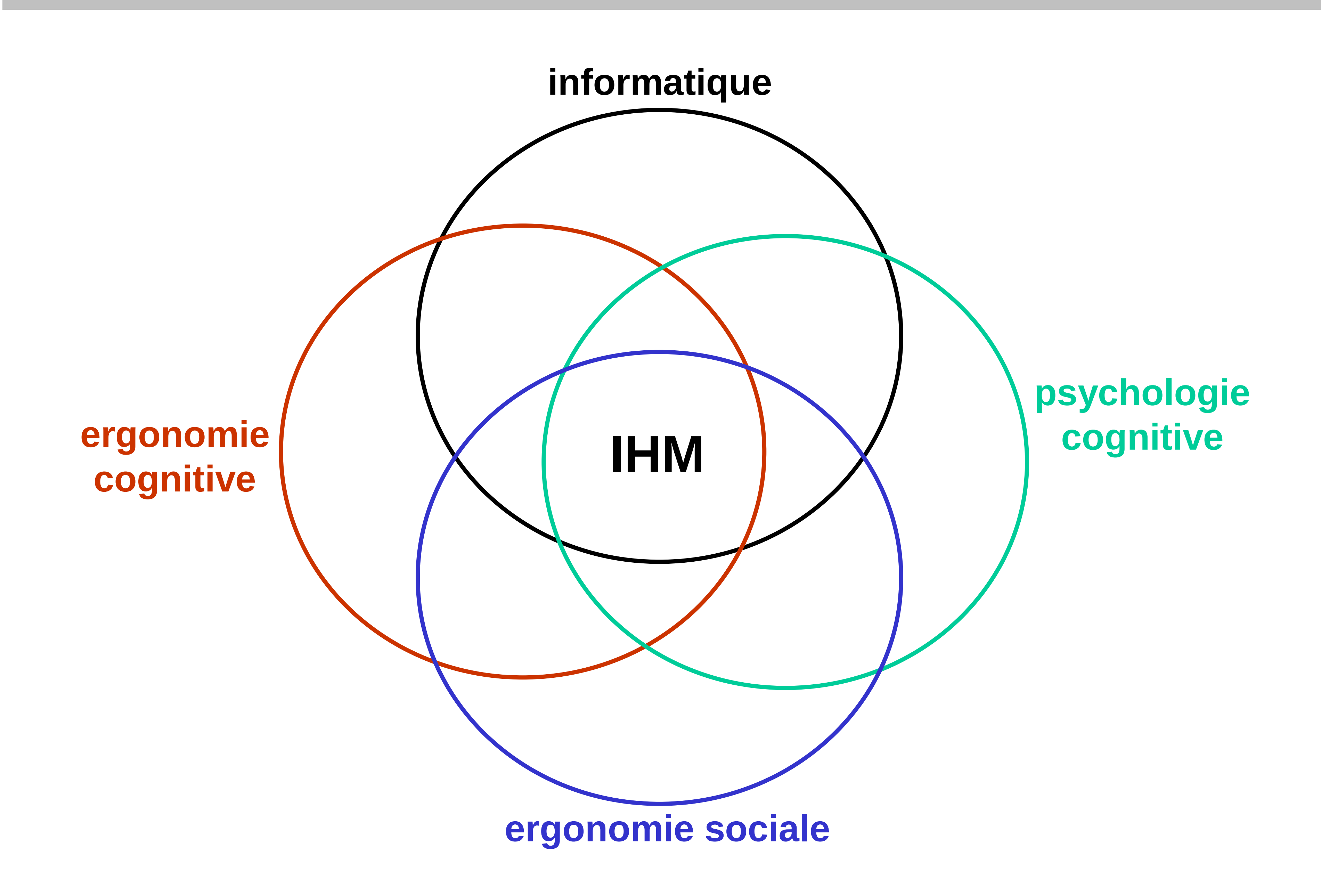
  ⑤ parties 1 à 9 : généralités et dispositifs matériels
  ⑤ parties 10 à 17 consacré au génie logiciel

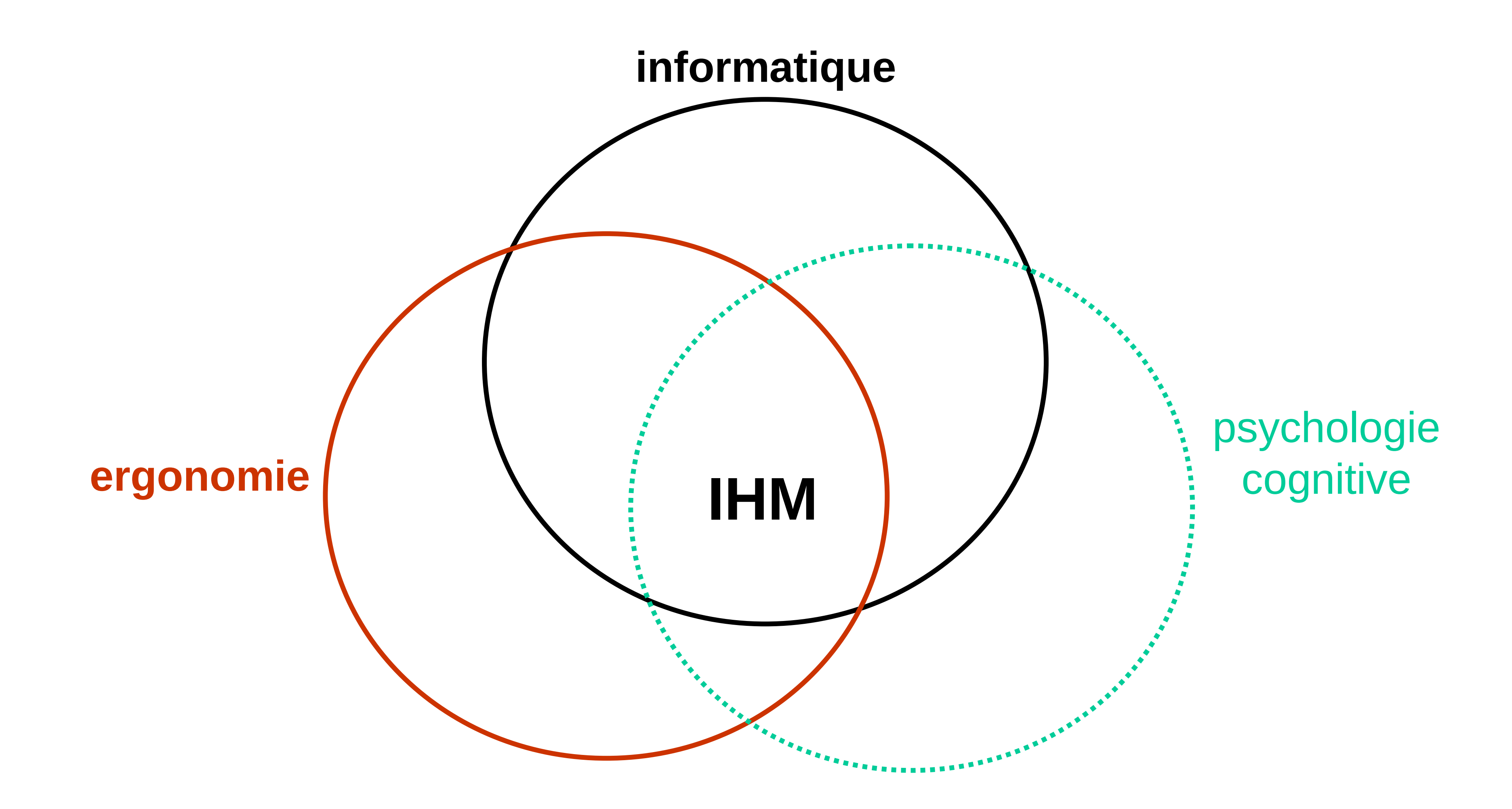
- ✓ ISO 14915 *Ergonomie des logiciels pour les interfaces utilisateur multimédias*

Normes AFNOR

- ✓ Z67 -110 *Ergonomie et conception du DHM. 1988.*
- ✓ Z67-133-1. *Evaluation des produits logiciels (Partie 1: définition des critères ergonomiques de conception et d'évaluation des interfaces utilisateurs*. 1991

IHM : APPROCHE PLURIDISCIPLINAIRE




informatique
ergonomie
IHM
psychologie
cognitive

IHM : RECOMMANDATIONS ERGONOMIQUES

Recommandations génériques à toute application interactive

- erreurs: gestion, prévention
- design (non) anthropomorphique
- affichage: alignement, couleurs, polices de caractères …

⑤ *chapitre 3*

Recommandations spécifiques aux interfaces WIMP

- Multi-fenêtrage
- Organisation de menus
- Manipulation directe : icônes et pointeur souris

⑤ *chapitre 4*

Recommandations spécifiques aux applications WWW

⑤ *chapitre 5*

BIBLIOGRAPHIE

Ouvrages généraux

Mayhew D.J. (1992) Principles and guidelines in software user interface design. Prentice-Hall

Cours sur la Toile

Giboin A. *Ergonomie des IHMs.* ESSI. www.essi.fr/~pinna/MODULEIHM

Travaux cités

Bury K. Davie S., Darnell M. (1985) *Windows Management: a review of issues and some results from user testing*, IBM Human Factor Center, research report HFC-53, San Jose, Ca,1985.

Nielsen J. (1994) Heuristic evaluation In Nielsen J. & Mack R.L. Usability Inspection Methods, John Wiley.

Norman D. (1998) *The design of everyday things*. MIT Press.

Guidelines - Normes

Apple Computer (1995) Macintosh Human Interface Guidelines, Addison-Wesley

Apple Computer (2002) *Aqua Human Interfaces Guidelines*., Addison-Wesley

Blanchard H. (1997) User interface standards in the ISO Ergonomics Technical Commitee, *SIGCHI Bulletin*, Vol. 29, N°1, 20-22

Brown C. M. (1988) *Human-Computer Interface Design Guidelines.* Xerox Corp.

Mayhew D. (1992) *Principles and guidelines in software user interface design*. Prentice Hall, Hemel Hempstead

Microsoft Corportation (2001) *Windows XP Visual Guidelines*.

Open Software Foundation (1991) *OSF/Motif Style Guide*. Prentice Hall, Hemel Hempstead.

Smith S., Mosier J. (1986) Guidelines for designing user interface software. Mitre Corporation report MTR-9420. Mitre Corporation.Bedford, MA

Sun Microsystems Inc. (2000) *Java Look and Feel design Guidelines.*

Vanderdonkt J. (1995) *Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur.